PART I

Foundations

Chapter 1

Convergence of Communities

e-Business is a catalyst for change. The change has been profound and fast as companies adopt the Internet as a primary channel for business interactions. Businesses initially used the Web as a one-way channel for communicating text and graphical information to consumers. The Web quickly became a platform for electronically processing orders and making catalog and real-time inventory available online. This consumer-oriented business transformation has now begun to impact the very heart of supply chain operations. Recent forecasts predict that business-to-business (B2B) transactions via the Internet will soon dwarf the other e-business conducted with consumers. Behind the scenes of this publicity, however, information-technology professionals are scrambling faster than ever. Yet, they are still falling behind in their efforts to interconnect all of the new and legacy systems required to fulfill this new electronic-business age.

e-Business requires integration of the information and processes needed to conduct business in real time. At the consumer level, this means that online catalogs must access the inventory database; also, credit authorization, order processing, and fulfillment must be integrated to deliver the goods to each buyer. Challenges have been encountered but were conquered, and the world moved on with this new, faster channel for sales. Then came B2B integration. Supply chain integration of manufacturers and distributors requires deeper introspection into sales forecasts, production scheduling, product configuration, and inventory management. The arrival of electronic marketplaces has created a "brave new world" of electronic bidding, auctions, reverse auctions, and a steady stream of never-before encountered business processes—all of which needs to be completed yesterday.

Finally, as if these challenges weren't difficult enough, all these new services must be delivered via personalized portals that can be accessed using Web browsers, personal digital assistants (PDAs), cell phones, pagers, interactive television, and automated shopping agents. These portals must become an extension of the core enterprise information infrastructure, not simply patched on as a Web sales channel. The portal becomes a secure conduit for basic operational data to be delivered to remote and mobile employees, key business partners, suppliers, and customers. The portal is a window into the B2B flow of information.

To satisfy these new demands, we must adopt a fundamental change in the way system integration is accomplished. This means an infrastructure that supports loose coupling of intra- and inter-enterprise information between widely disparate application designs, operating systems, databases, and application programming interfaces (APIs). The eXtensible Markup Language (XML) has become a solution for many of these needs. XML is not a magic wand that can solve all problems, but it does enable us to focus on the definition of shared vocabularies for exchanging information that can be processed easily by both human and computer systems. XML and its domain vocabularies are becoming the *lingua franca* for B2B communication.

An additional benefit of XML is that it was derived from a document-processing heritage for supporting both computer and human communication. As a result, it contains standardized stylesheet processing languages and tools for presenting XML documents to human users in many formats—print, multimedia, and synthesized voice. Through these technologies, XML has the potential to become the standard platform for convergence of information to all types of portals.

A complete coverage of how XML can address these e-business issues would require many volumes. This book adopts a more modest goal of addressing business-to-business vocabularies and portals using XML. Our main goal is to describe the use of the Unified Modeling Language (UML) as a technique for designing business vocabularies that can be deployed using XML for e-business integration.

Models for e-Business

By itself, XML is only a syntax for the exchange of data and text-oriented documents; yet we need more than a common syntax for successful communication. *Communication* requires shared models of both the underlying domain semantics, and the processes and policies used to engage in electronic commerce. These models are the very essence of B2B integration. They may be implicit in the applications that process the XML documents; or they may be explicit in definition of the model's concepts, relationships, and constraints. In practice, the models are defined both implicitly and explicitly.

We'll cover the following three aspects of modeling B2B communication.

- *Modeling system requirements with use cases.* These models define the roles of stakeholders and the use case actors who are involved in B2B interactions, plus the functional requirements of those stakeholders and actors.
- *Modeling processes and communication policies.* B2B interactions are not limited to sending a single message but require a coordinated sequence of activities and expectations of the business partners.
- *Modeling business vocabularies.* Each message exchanged within a communication process contains information content that may be short and simple or very long and complex. Each of these content documents is defined by a vocabulary that is shared by the parties engaged in the communication.

XML is becoming widely used for representing both the process and the content information when deploying models. Process information includes the messaging infrastructure and workflow control that guides the process execution. Many B2B processes are asynchronous and long running, so the XML-based message header information identifies the parties, process, and purpose of the message. The business vocabularies define the heart of the message—its content. Example product catalog vocabulary is developed in this book. The catalog data using this vocabulary will be exchanged in messages between business partners when aggregating catalogs from multiple suppliers or when responding to queries for product specification data.

An XML application is, however, much more than structured data! The application is part of a broader system context, including both architectural and process requirements. Most e-business applications contain requirements, from both business and technical stakeholders, which are distributed across an inter-enterprise system. Development of these systems benefits greatly from visual models and a process that encourages active communication. Let's face it—the business world revolves around graphical presentation, so anything that adds a visual component to XML specification is very helpful.

Stakeholder Communities

In the beginning stages of developing an e-business system, many stakeholders must contribute requirements, domain model details for the analysis, and specifications for design and implementation. This book does not attempt to cover the full development lifecycle but instead focuses on analysis and some parts of design, all with an eye toward the use of XML. Within this context, we must consider the needs of several diverse stakeholder communities. I have narrowed the focus for this book to five groups: consumers, business analysts, Web application specialists, system integration specialists, and content developers.

A complete analysis of an e-business system would require documenting many different use cases that describe the convergence of these stakeholder communities in B2B and business-to-customer (B2C) interaction scenarios. Figure 1-1 illustrates one high-level use case diagram that incorporates all five of the identified stakeholder communities. This high-level view spans the requirements from designing the product catalog structure and content to presenting personalized views of this catalog to consumers.

A use case specification identifies a set of "actors" that interact with each use case. Each actor represents the role of a user that works with the resulting system after it is deployed; a specific human user may fulfill several different roles. As shown in Figure 1-1, UML use case diagrams always represent an actor using a "stick person" icon. Each oval represents an individual use case that describes (in a separate document and/or diagram) the activity flow required to achieve the goal of that use case. The use case diagram presents an outside view (from the perspective of the actors) of the system without specifying *how* that functionality is designed or implemented.

I use the term *stakeholder* rather than actor in this description because of its broader meaning. A stakeholder represents all roles that have an interest in the goals of a use case, not only those who directly participate in performing the use case activities. Thus, a Consumer stakeholder provides requirements for the design of portals and business processes, although he or she may not have a direct role in performing the steps within those use cases.

Future chapters will decompose the individual use cases into scenarios illustrated with class diagrams, object diagrams, sequence diagrams, and activity diagrams that specify how XML fits into the overall e-business solution. But first, I'll describe the typical characteristics of each stakeholder community in a bit more detail.

Consumer

A consumer may be either a business customer in the supply chain (for example, a buyer of components for a manufacturer) or an end-user retail customer. I use the term *consumer* rather than customer to emphasize that consumption occurs at many points along the e-business supply chain. In many current situations, this consumer represents a human user who is interacting with the product catalog service. However, we will increasingly see automated consumer

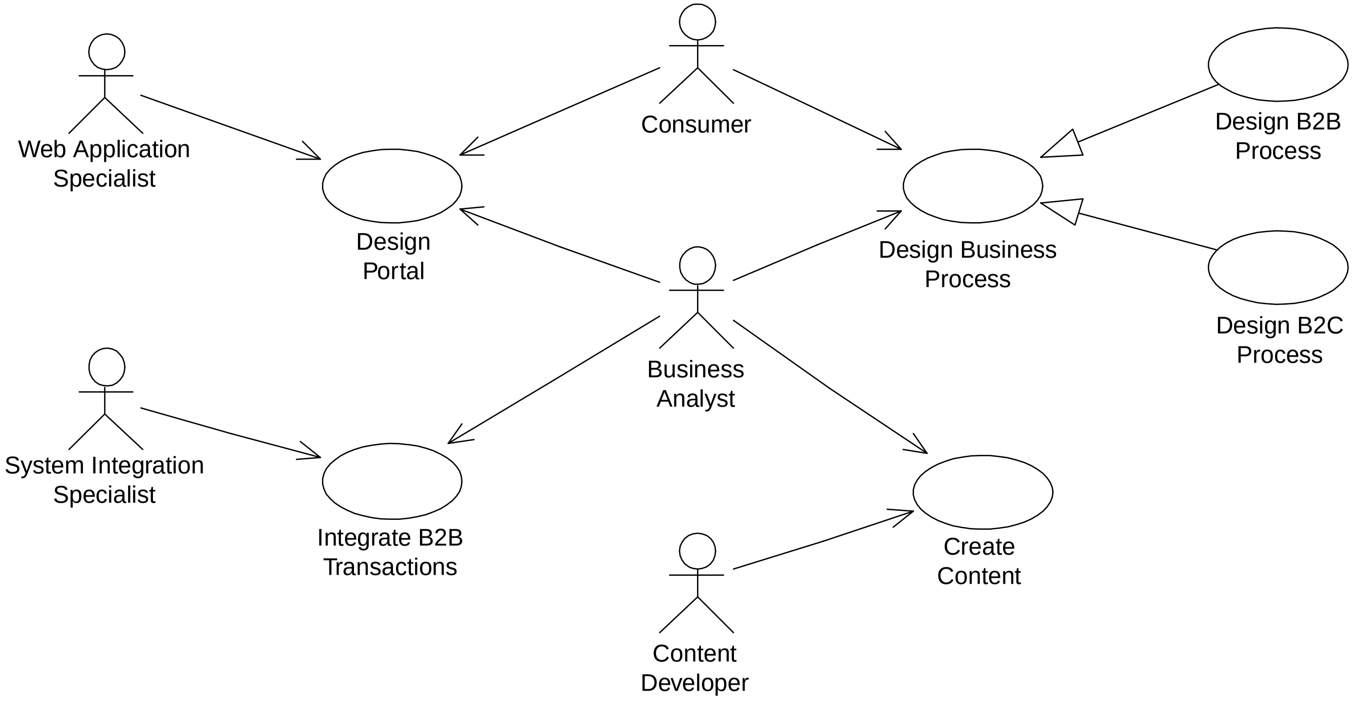

Figure 1-1 e-Business stakeholder interactions

agents, also called 'bots (a truncated form of the word robot), that fulfill the consumer role. A B2B consumer may be an automated order submission process triggered by the production schedule, or a retail consumer may be represented by an automated shopping agent that performs comparison shopping across sites.

The use case diagram shows the consumer as a stakeholder in both designing the business processes and in designing the portal's personalized views of the catalog. The business processes for business consumers are very different from those supporting retail customers. Therefore, the use case diagram includes two more specialized use cases that extend the general use case for designing business processes. During requirements analysis of a B2B service, representative consumers must be included, either in individual interviews or as part of larger focus groups. This input would be documented in text documents associated with the relevant use cases and in activity or sequence diagrams of the event flows.

The difference in requirements for human or 'bot consumers is unspecified in this high-level use case diagram. Their different requirements would need to be described in the individual use cases, or likely in additional sub-use cases that define the specialization. For example, both a human and an automated 'bot consumer require a personalized presentation of the catalog, but the means of personalization may be quite different. A human user would expect an HTML browser-based presentation that is tailored to his or her preferences and supports quick navigation without a lot of distracting, irrelevant information. On the other hand, a 'bot may expect a filtered subset of the catalog content that is tailored to its particular interests as a buyer. Of course, either of these kinds of personalization can be enabled by effective use of XML.

In the context of an XML application, consumers are both sources and recipients of communication. As a source, the consumer produces XML-based messages containing product catalog queries, requests for quotes, or orders. A consumer receives product catalog information encoded in XML documents, which is then presented in a Web browser or imported into the recipient's business system. Both scenarios are covered in later chapters.

Business Analyst

The use case diagram in Figure 1-1 depicts the business analyst as being central to implementing a successful e-business application. The analyst participates in each of the use cases, often as the integrative force and visionary for the overall system. The business analyst also collaborates with the content developer to create the domain model for the product catalog. Other stakeholders such as marketing, sales, and accounting would be included in a full requirements analysis. The analyst then works with the Web application specialist to design a dynamic, personalized Web site for this product catalog.

The business analyst uses this knowledge of the product catalog model to collaborate with the system integration specialists. System integration requirements include adaptation of legacy systems into this new e-business application at multiple stages of both B2B and B2C business processes, plus integration with other external systems such as credit authorization and shipping.

The business analyst uses his or her knowledge of the overall e-business environment to oversee the requirements gathering process for an integrated solution. He or she must understand the process for using the UML in use case and activity modeling and be able to facilitate design of the UML static structure models that represent the XML vocabularies for system integration and Web presentation. This last modeling responsibility is the central topic of this book.

Web Application Specialist

A Catalog Exchange Service (CatX) is analyzed as a sample application in this book. For the CatX system, Web application specialists are most often concerned with the portal server component and portal requirements and design. But this is no simple matter! We are still very early in the maturity of XML processing directly within Web browsers. So for the next year or two, it's likely that much of the Web application processing for XML content will be executed within application servers. For example, Java servlets may process the XML catalog content, using an XSLT transformation to produce HTML content for the e-marketplace portal interface.

There are many sub-roles within the community of Web application specialists. Dynamic server pages have become the mainstays of Web development; for example, the J2EE standard for JavaServer Pages (JSP), Microsoft's Active Server Pages (ASP), or Allaire's ColdFusion. These technologies are often used to process XML content for Web browser delivery. However, XML is also introducing new technologies into the Web developer's toolkit. The Extensible Stylesheet Language (XSL) is now used in combination with Cascading Style Sheets (CSS) to transform and format XML documents for presentation on the Web. These topics are discussed in later chapters as part of designing XML portals.

Jim Conallen's book [Conallen, 2000] examines the broader details and alternative architectures for designing Web applications. I focus attention on the analysis and design of XML content within e-business solutions. Chapters 2 and 5 explore the use of XML by Web application specialists and introduce the benefits of UML from their perspective.

There are also many exciting new developments in wireless Internet access via mobile phones, PDAs, and other specialized Internet devices. In most cases, XML plays a significant, if not dominant, role in communicating information to and from these wireless devices. This new stakeholder role fits most closely with the skills of advanced Web application specialists but will also require development of new skills. I'll have more to say about this subject in later chapters, particularly regarding the use of XSLT to transform XML catalog content for presentation by the wireless world.

System Integration Specialist

System integration specialists play a very significant role in B2B e-commerce. It is almost a certainty that one will encounter several incompatible systems when integrating e-business processes across several organizations in a supply chain or marketplace. Prior to the availability of XML, the most common solutions were to attempt development of standard APIs using technologies like CORBA, RMI, or DCOM, or to adopt message-oriented middleware that supported reliable, asynchronous routing and delivery of data and events. These approaches proved difficult and expensive to implement in a widely distributed, heterogeneous environment using the public Internet for transport and did not have a good solution for formatting the large blocks of structured data contained in the message's body.

XML-enabled applications take a very different approach to integration. Instead of standardizing the APIs for system integration, the data format is standardized for text-based documents exchanged among participants. This is sometimes referred to as platform-independent data, which complements the Java programming language as platform-independent code for a complete, vendor-independent integration solution.

Use of XML for the content format allows these documents to be parsed and transformed with the use of standards-based tools that are very inexpensive, or often free. Thus, exporting and importing XML documents has become a common means for integrating otherwise incompatible systems. Many B2B integration solutions are now adopting a hybrid of message-oriented middleware, carrying XML structured content in the message body.

These are all primary tasks within the domain of system integration specialists. The use case diagram shown in Figure 1-1 includes one use case for integrating B2B transactions. Obviously, there are many details yet to be explored in these requirements. Chapters 3 and 4 introduce the use of UML for modeling business processes and the XML vocabularies required for B2B integration.

Content Developer

Within a large organization, hundreds of people may be involved in developing catalog content in the form of product specifications, price lists, white papers, frequently asked question (FAQ) lists, and so on. Since all of this content is potentially useful in a B2B service, the content developers must be involved in specification of the catalog vocabulary. These stakeholders will be consulted in upcoming chapters as we construct the UML models for the required behavior and structure of the XML documents.

Road Map for This Book

Figure 1-2 presents a road map for the topics in this book, and the following sections explain the contents of the three parts of the book. Each chapter concludes with a list of steps for success. The purpose of these steps is to distill general principles for modeling XML applications and to organize them along the guidelines of the Unified Process. Consider these steps as a supplement to rather than a replacement of the Unified Process. Their goal is to give you a road map that helps achieve successful completion of your own XML applications.

Part I. Foundations

Chapter 2 introduces XML by comparing it with a similar application of HTML. The example, using the Rich Site Summary (RSS) vocabulary for news headlines, builds a foundation for using the same vocabulary in future chapters. The chapter covers the center part of Figure 1-2 by describing XML documents and Document Type Definition (DTD) schemas, plus use of XSLT to transform XML into HTML for presentation.

Chapter 3 introduces UML models, thus covering the upper part of the road map. Each of the UML diagrams is described based on its role in an iterative development process using the RSS example from Chapter 2.

Chapter 4 analyzes the use case requirements for e-business integration. This chapter introduces the Catalog Markup Language (CatML) that is used throughout the remainder of the book. CatML was conceived and implemented strictly for the purposes of this book.

Chapter 5 analyzes the use case requirements for wired and wireless portals. “MyCat” is introduced as a personalized portal for our catalog exchange service example.

Part II. XML Vocabularies

Part II focuses entirely on the analysis and design of XML vocabularies using UML class diagrams. All examples are based on the CatML vocabulary.

Chapter 6 describes XML vocabulary design and works through seven design issues that must be considered when mapping UML class models to XML. UML object diagrams are used to describe the mapping scenarios.

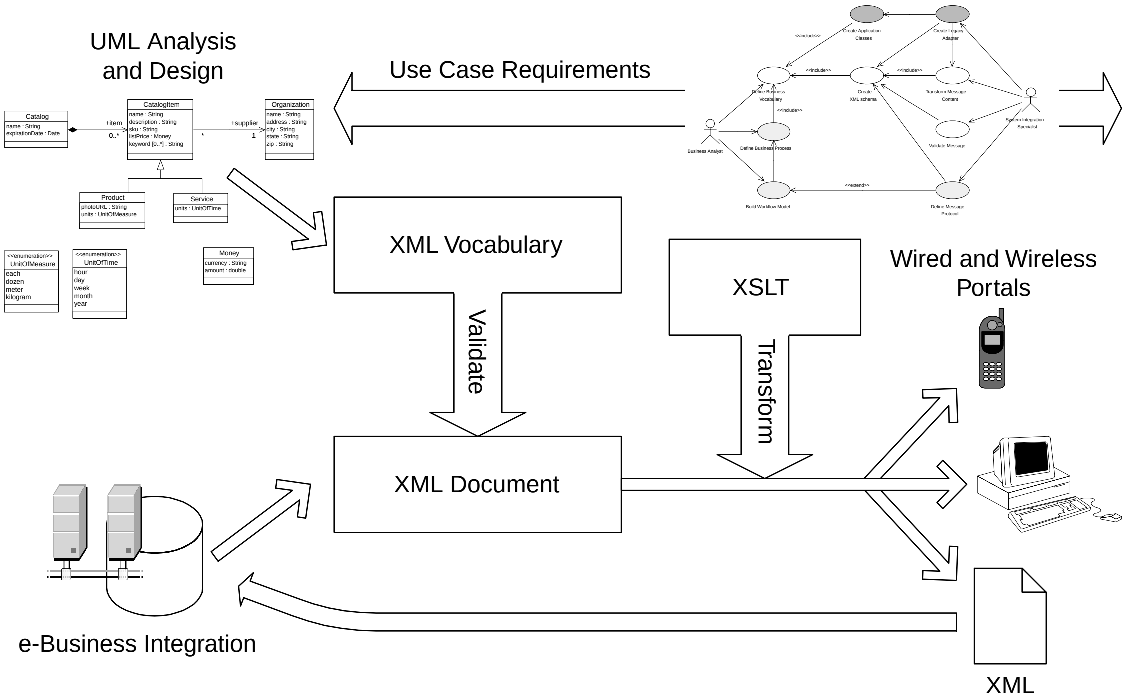

Figure 1-2 Components of an e-Business application

Chapter 7 takes the next step into distributed applications and describes four XML linking mechanisms used to map UML relationships into XML hyperlinks. This chapter completes our design specification for the CatML vocabulary.

Chapter 8 provides a detailed introduction and comparison of XML DTDs and Schemas.

Chapter 9 describes the criteria and production rules used to generate XML DTDs and Schemas from UML class diagrams. Alternative rules and tradeoffs are analyzed for producing relaxed versus strict schemas. All examples are based on generating schemas for CatML.

Part III. Deployment

Chapter 10 returns to the e-business integration issues introduced in Chapter 4. Two detailed examples are described using XSLT to transform the CatML vocabulary to and from the RosettaNet standard XML vocabulary.

Chapter 11 probes deeper into deployment of the portal that was introduced in Chapter 5. XSLT is again used to transform CatML, but this time into HTML for presentation in the Apache Jetspeed portal framework.

Chapter 12 outlines the common features of current e-business architectures for deploying B2B applications. As part of the architecture, the Simple Object Access Protocol (SOAP) is introduced as an XML-based messaging standard for interconnecting distributed components.

Steps for Success

1. Identify e-business drivers. What strategies of your business are driving the need for integration? What are your customers' and suppliers' strategies? Where are your competitors headed?
2. Identify stakeholder communities. A successful XML application must consider the requirements of all major stakeholder communities, including consumers, suppliers, business analysts, system integrator specialists, Web application specialists, and content developers.
3. Create a high-level use case diagram for your system context.
4. For each use case, identify its primary goals and assumptions.
5. Identify an initial set of workflow processes among the stakeholders.